Fraud Analytics Assignment 1

Ananthoju Pranav Sai
ai20btech11004

Dishank Jain
ai20btech11011

Shivangana Rawat
cs20mtech12001

Kalvakuntla Umesh
cs20btech11024

Nyalapogula Manaswini
cs20btech11035

1 Problem Statement

Circular trading also known as round-trip trading, is a form of tax evasion in Goods and Services Tax where a group of fraudulent taxpayers aim to mask illegal transactions by superimposing several fictitious transactions among themselves in a short period. We try to detect circular trading using node2vec and DBSCAN algorithms.

2 Dataset

We are given a dataset consisting of 130535 invoices. The data consists of 3 columns Seller ID, Buyer ID, and Value.

	Seller ID	Buyer ID	Value
count	130535.000000	130535.000000	1.305350e+05
mean	1309.358287	1182.851258	6.930965e+05
std	294.435026	169.809657	5.696676e+05
min	1001.000000	1001.000000	1.000600e+04
25%	1078.000000	1060.000000	2.361085e+05
50%	1214.000000	1112.000000	5.571960e+05
75%	1488.000000	1276.000000	1.074405e+06
max	2190.000000	1887.000000	2.124000e+07

Figure 1: Data description

3 Algorithms

3.1 Converting MulitDiGraph to Undirected Graph

For all the given invoices we use networkx package to construct a Directed Multi Graph where the edge from u to v with weight w in the graph represents an invoice of amount w from seller u to buyer v. After constructing the directed multigraph, we need to convert it to an edge-weighted undirected simple graph. The weight of an edge xy in this graph is directly proportional to the number of three cycles and two cycles containing both x and y. [2] mentions the importance of 3-cycle weighting. We used a similar approach to find the weights of the edges in the undirected graph.

Algorithm 1 Converting Directed Multigraph to Undirected Graph

```
G : MultiDiGraph
UG: Graph
for u ∈ G.nodes() do
    for v ∈ G.successors(u) do
        w = 1
        if Edge(u-v) is in directed 3-cycle then
            w = 2
            if the 3-cycle has one reciprocal edge then
                w= 3
            else if the 3-cycle has two reciprocal edges then
                w= 3
            else if the 3-cycle has three reciprocal edges then
                w= 4
            end if
        end if
    end for
    if UG.has_egde(u,v) then
        UG[u][v][weight] = max(UG[u][v][weight],w)
    else
        UG.add_edge(u,v,weight=w)
    end if
end for
```

3.2 Node2Vec

After obtaining the undirected graph. We use Node2Vec to learn the node embeddings. The Node2Vec [1] algorithm is based on the skip-gram model used in natural language processing. It majorly involves two steps.

- Generating random walks on the graph: Random walks are sequences of nodes that are obtained by starting at a particular node and then taking

successive steps to neighbor nodes using a random policy. It uses a biased random walk procedure that balances exploring different neighborhoods and preserving the structural information of the graph

- Learning node embeddings: Node embeddings are low-dimensional vector representations of the nodes in the graph. Node2Vec uses a neural network to learn these embeddings from the random walks generated in the previous step. We use a neural network that takes input as the context nodes (i.e., nodes that occur within a window size proximity in the random walk) and predicts the current node.

The Node2Vec algorithm is able to capture the structural information of the graph by balancing local and global information. We use the node2vec framework to implement the algorithm.

3.3 DBSCAN

The learned node embeddings are then used by DBSCAN (Density-Based Spatial Clustering of Application with Noise). DBSCAN [3] is a popular density-based clustering algorithm that is used for finding clusters. There are two main parameters in DBSCAN

- Epsilon (ϵ): The maximum valid distance between two points of the same cluster.
- MinPts: The minimum number of points required to form a dense region or cluster.

DBSCAN has several advantages over other clustering algorithms such as k-means and hierarchical clustering, especially when dealing with complex and non-linear data structures. However, it also has some limitations, such as the sensitivity to the choice of parameters and the inability to handle clusters of varying densities. We use scikit-learn's implementation of DBSCAN.

3.4 Principal Component Analysis

After clusters are obtained and labeled, we use PCA to reduce the embedding dimensions to two and plot the obtained clusters. The main idea behind PCA is to find a new set of orthogonal variables, called principal components, that capture most of the variation in the original data. We use scikit-learn's implementation of PCA.

4 Results

We plot the obtained clusters after PCA. We can see that other than the blue-colored nodes, the remaining ones are involved in circular trading.

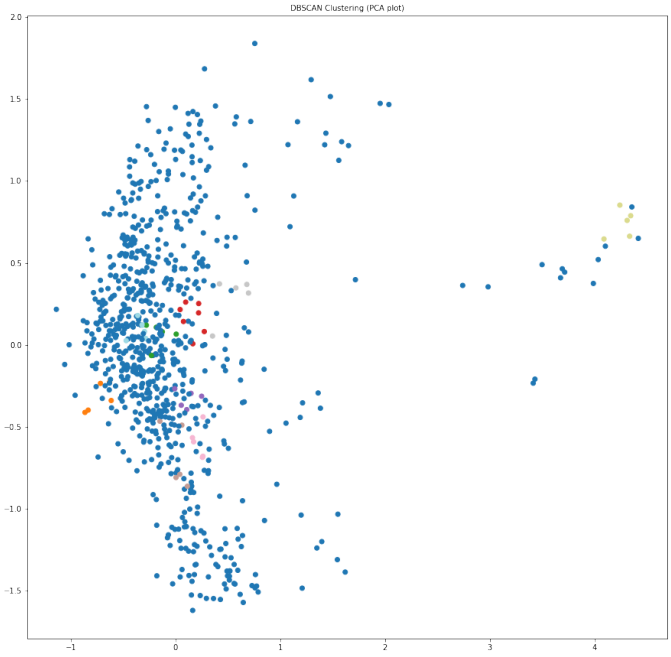

Figure 2: PCA plot of clusters

The graphs formed by the sets of nodes that are involved in circular trading are shown below.

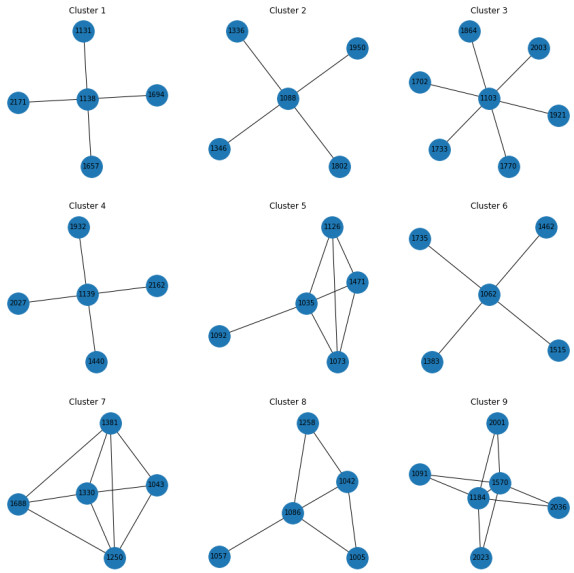

Figure 3: Detected circular trades

5 Conclusion

We constructed a directed multigraph from the given dataset and converted it to an undirected graph. We then used the node2vec algorithm and DBSCAN algorithm to detect circular trading in the given dataset.

References

[1] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. 2016. arXiv: `1607.00653 [cs.SI]`.

[2] Christine Klymko, David Gleich, and Tamara G. Kolda. *Using Triangles to Improve Community Detection in Directed Networks*. 2014. arXiv: `1404.5874 [cs.SI]`.

[3] Jörg Sander et al. "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". In: *Data Mining and Knowledge Discovery* 2 (1998), pp. 169–194.